

Rapid Geometry Creation for Computer-Aided Engineering Parametric Analyses: A Case Study Using ComGeom2 for Launch Abort System Design

Veronica Hawke¹, Peter Gage², and Ted Manning³
ELORET, Corp., Sunnyvale, CA 94087

ComGeom2, a tool developed to generate *Common Geometry* representation for multidisciplinary analysis, has been used to create a large set of geometries for use in a design study requiring analysis by two computational codes. This paper describes the process used to generate the large number of configurations and suggests ways to further automate the process and make it more efficient for future studies. The design geometry for this study is the launch abort system of the NASA Crew Launch Vehicle.

1. Introduction

Design of aerospace vehicles requires multidisciplinary analysis in such diverse areas as aerodynamics, heat transfer, structural dynamics, weight estimation, cost analysis, and safety. For a rapid and efficient design process, it is important that analysis codes be well integrated. Ideally, the vehicle design is represented in a common fashion across codes so that analysis results are readily transferable and consistent as updates to vehicle geometry are made. It is also important that the degree of detail carried in the model matches the fidelity requirements of the analysis in question. However, an application for a given analysis discipline is traditionally developed in isolation of other disciplines, resulting in a unique representation of vehicle geometry. Such non-standard representations require extensive model and data translation.

ComGeom2 (for Common Geometry, version 2) is a computer program that permits convenient customization and assembly of pre-defined, parametrically driven CAD components. It is a generalization of the original ComGeom [1], which worked with a predefined assembly of parts, and did not permit construction of new assemblies. Intended for computer aided engineering (CAE) analysts who need precise control of geometry and boundary conditions, the software delivers:

- A collection of properly sized CAD parts;
- A surface mesh or geometry suitable for direct use in analysis or as the basis of a volume mesh;
- Mesh-based surface tags to track components and user-defined regions; and
- Geometric measurements.

The strengths of ComGeom2 lie in its tight integration with CAD, ease of system model configuration, and flexible handling of user-defined tagged regions. It directly interrogates, modifies, and accurately meshes Pro/ENGINEER [2] solid geometry (and is readily extensible to other CAD kernels). The application's automatic control of parametric geometry facilitates iterative analysis, including design of experiments (DOE), shape sensitivity studies, and physics-based shape optimization.

This paper focuses on a particular application of ComGeom2 in an engineering trade study. While the motivations for ComGeom2 and its features have been reported previously [3], its impact on a rapid design cycle needed to be assessed. Here we consider requirements imposed on the CAD parts to be used with ComGeom2 and evaluate the incremental effort associated with this customized part development. The design study used for this evaluation, which is concerned with arrangement of rockets in the Launch Abort System (LAS) of the NASA Crew Launch Vehicle (CLV), is described in Section 2. Discussion of the

¹ Research Scientist, AIAA Member

² Technical Director, AIAA Associate Fellow

³ Senior Research Scientist, AIAA Member

relevant geometric features appears in Section 3, and development of CAD parts for ComGeom2 is described in Section 4. Requirements for integration of ComGeom2 into the ModelCenter problem-solving environment [4], which facilitates execution of trade studies, are discussed in Section 5, while implementation of this trade study is reported in Section 6. Execution efficiency of the integrated system as well as the method used for quality checking of the output is evaluated in Section 7. The paper concludes with a summary of findings and some observations on opportunities for future performance improvements.

2. Launch Abort System Design

The work we describe in this paper contributed to ongoing aerodynamics studies that support the design and development of the Ares I Crew Launch Vehicle (CLV) under the auspices of the Exploration Launch Office of the NASA Exploration Systems Mission Directorate (ESMD)[5]. The CLV will be used to launch astronauts to low earth orbit and rendezvous with either the International Space Station or the ESMD's earth departure stage for lunar or other future missions beyond low Earth orbit. The Orion Crew Exploration Vehicle (CEV), which is made up of the Launch Abort System (LAS), the Crew Module (CM), and the Service Module (SM), forms the upper stage of the CLV (Figure 1). With its tower, canted nozzles, and boost protective cover, the LAS will dominate CLV ascent drag. The need to understand the influence of the geometric features of the LAS on launch drag motivated a geometric trade study.

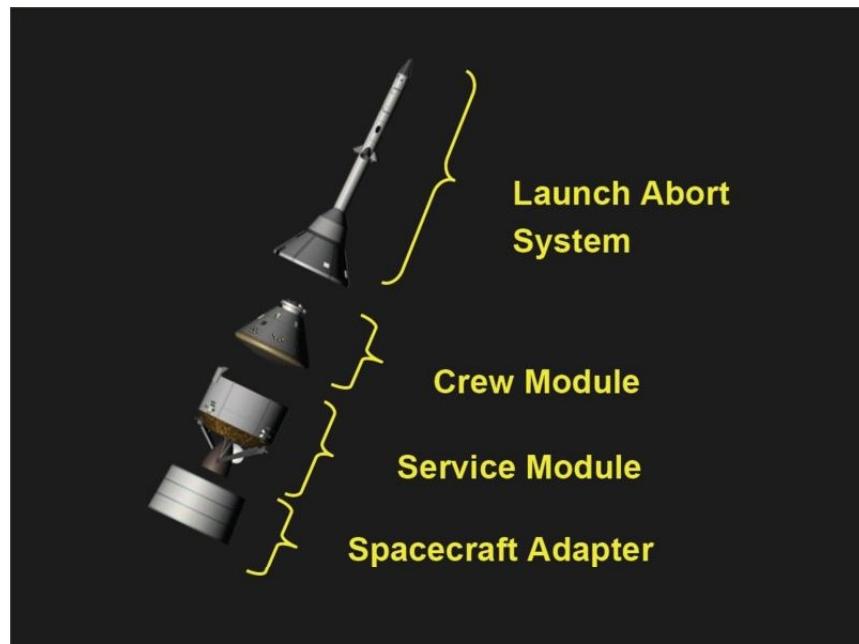


Figure 1: Orion Crew Exploration Vehicle components. (Source: NASA)

Engineers at NASA are designing the LAS to crew escape in the early phases of ascent. The LAS will be similar to escape systems used in previous crewed launch vehicles, including Mercury, Gemini, and Apollo [6]. Figure 2 illustrates the assembly of tower and CM at the top of the launch stack. In the event of a booster failure, solid rockets in the tower fire to pull the CM away from the booster. The abort system must ensure that the CM reaches an altitude and orientation that permits safe parachute deployment, and that the CM has adequate lateral separation from the boosters.

For Apollo, a boost protective cover was added to the Command Module to protect its surface coating from contamination during the boost phase [7]. The cover also protects the crew from the high-temperature exhaust from the LAS rockets in the event of an abort. There is an opportunity to modify the LAS design for the current vehicle so that plume impingement on the CM is avoided.

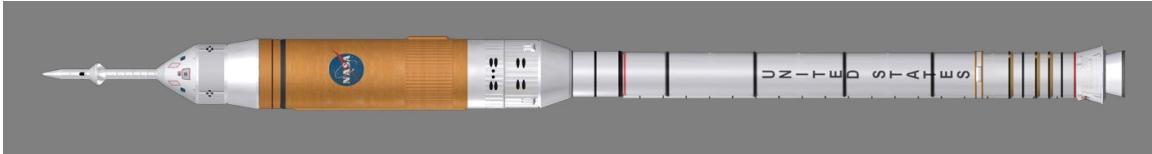


Figure 2: The Launch Abort System Tower Attached to the Crew Launch Vehicle.
(Source: NASA)

This geometric study is motivated by the need to evaluate reverse flow nozzles as an alternative to nozzles lower down on the structure. Nozzles placed too low might have a plume that impinges on the shroud causing higher pressure and heat loads. The influence of the location and size of the nozzle housing (the “flare”) on thrust (due to nozzle orientation) and on aerodynamic drag (due to shock structure) will be evaluated in a trade study. The nose shape of the LAS also needs to be optimized to achieve drag reduction.

3. Requirements for Launch Abort System Geometric Trade Study

The Ares I CLV Aerodynamics effort encompasses projects utilizing wind tunnel testing and computational fluid dynamics (CFD) methodologies [5]. While initial phases focused on evaluating the reference CLV outer mold lines obtained in the ESMD’s Exploration System’s Architecture Study (ESAS) [8], the work that followed reflected a lack of confidence that the proposed LAS shape was suitable. ESMD commissioned a LAS geometry trade study to better understand the dependence of CLV ascent performance on the details of the LAS geometry. The launch tower is understood to serve as an aerospike, a feature that is commonly used to reduce the drag of blunt-nosed missiles. The study involved both CFD and wind tunnel testing: CFD would be used to study a wide range of geometries, whereas the wind-tunnel phase would examine a promising subset of those cases. In this section we describe the computational geometry requirements for the CFD phase of the LAS geometric trade study. The two-week time limit on this work was also an important constraint.

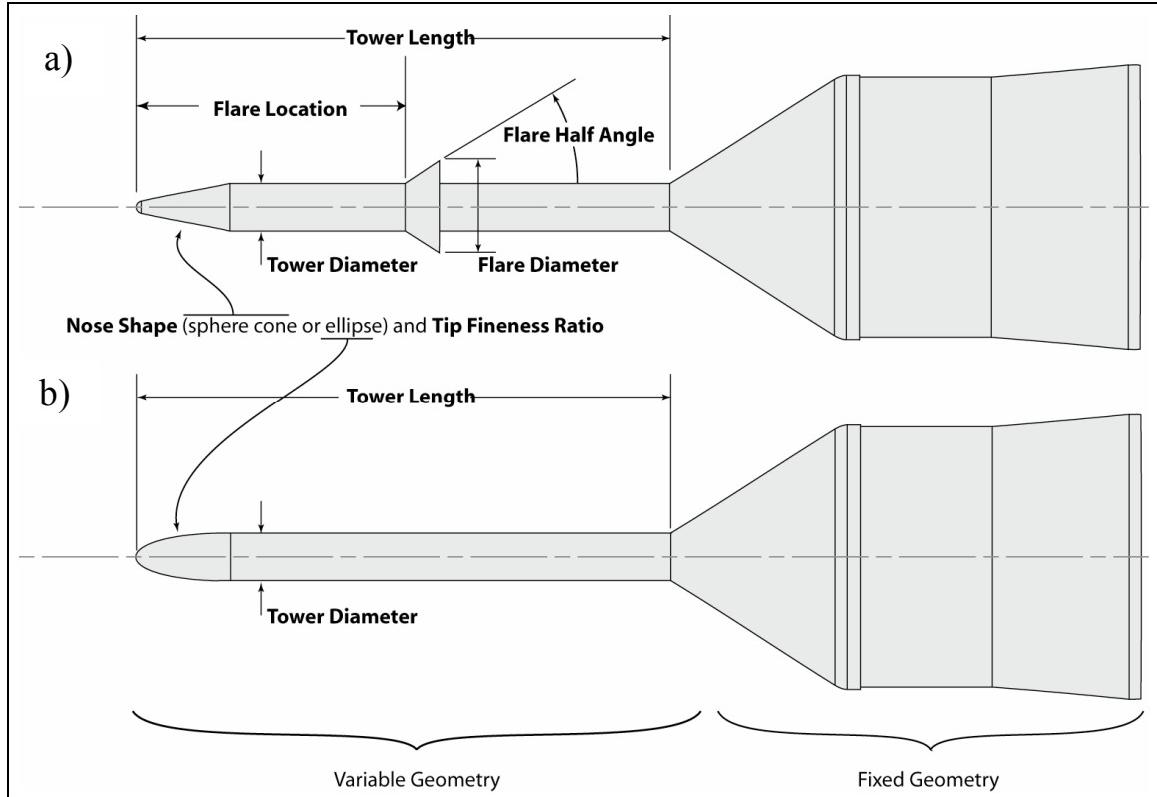


Figure 3: LAS geometry parameters: a) with LAS flare (7 parameters);
b) without LAS flare (4 parameters)

The primary goal of the trade study [5] was to determine which of the several geometric parameters defining the LAS impart the greatest influence on the forebody axial force coefficient at zero angle of attack. As depicted in Figure 3, the geometries required in this study were comprised of the LAS tower, the boost protective cover, the service module, and the second stage adapter. They are axisymmetric. Two variants were initially requested: tower with flare, and tower without flare. After the initial study, a variant without any tower was added.

The LAS forebody shape is driven primarily by tower diameter and length and nose-tip shape and fineness ratio. The alternative configuration adds a flare for shielding the LAS motor nozzles and inducing a drag-reducing shock ahead of the boost protective cover. The effects of the actual nozzle geometry was considered secondary to the effects of the flare and so were not modeled. The inclusion of the flare adds three parameters: flare location, diameter, and half-angle. The boost protective cover, service module, and second stage adapter were specified as fixed geometry, with the aft end of the adapter fixed in the streamwise coordinate ($x_{\text{pos}} = 1000 \text{ in}$). The study required that computational surfaces for the LAS geometries be supplied at three values in every parameter, which resulted in over 1500 unique configurations.

To reduce project risk in light of the trade study's time constraints, those managing the CFD effort divided the computational analysis among two flow solvers, Cart3D [9] and Overflow [10]. The study required that the geometry be supplied in a format compatible to these codes. Due to its speed, Cart3D, which solves the Euler equations on Cartesian grids, was designated for analyzing all 1500+ configurations at "low fidelity". Overflow, which solves the Reynolds-averaged Navier Stokes equations on structured overset-grids, was designated for analyzing a subset of these cases at "high fidelity", specially selected in a design of experiments (DOE) framework. Because each geometry instance is subjected to flow conditions covering a range of Mach numbers and angle of attack, the number of computational runs far exceeds the number of geometries,

Both CFD tools require a manifold computational surface as a foundation for creating volume grids. The analysts running Cart3D requested that these surfaces be supplied as triangulated surface meshes, whereas the analyst running Chimera Grid Tools [11] scripts to generate grids for Overflow requested that the surfaces be supplied as Pro/ENGINEER CAD parts. In the latter case, the parts were to each contain the features necessary to construct a generator curve, which the analyst's script revolves through 360° to reproduce each axisymmetric LAS body. The Cart3D surface mesh tolerances were not dictated at the outset but instead arrived upon iteratively, as described in section 7.4.

4. ComGeom2 Support for Computer-Aided Engineering Feature Requirements

The ComGeom2 feature-set satisfies the computer aided engineering requirements of the LAS study, which we discussed in the previous section. The authors' familiarity with ComGeom2 was also a factor in its selection.

ComGeom2 is, at its core, an integration tool that draws on a variety of software components to transform pre-defined CAD models and parameters into the computational geometry required by analysis. Central to ComGeom2's operation is the geometry kernel of Pro/ENGINEER. ComGeom2 manages parametric geometry in this kernel through two levels of application programming interfaces (APIs): CAPRI, and through CAPRI, Pro/TOOLKIT. CAPRI is a geometry abstraction layer that increases application portability [12]. Pro/TOOLKIT is Pro/ENGINEER's native API for custom applications. The ComGeom2 features we exploited are the standard CAD operations, such as parameter setting and model regeneration, and the CAPRI surface mesh tools. The CAPRI mesh tools efficiently represent a given CAD surface by distributing the mesh along mapping parameter directions. ComGeom2 is readily integrated into automated environments because it is command line driven and makes use of standard input and output file formats.

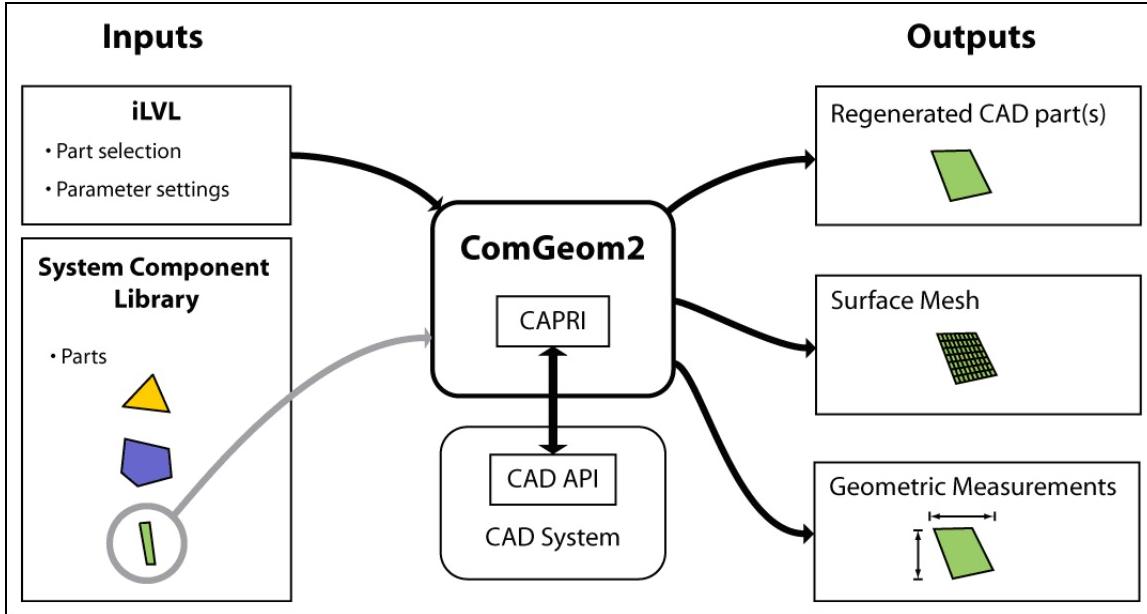


Figure 4: ComGeom2 data flow.

ComGeom2 data flow is depicted in Figure 4. Following part selections given in an “iLVL” input file (defined below), ComGeom2 first retrieves a number of CAD parts from a System Component Library (SCL). ComGeom2 then works with CAPRI and the attached CAD system to regenerate, mesh, and measure each part according to parameter settings appearing in the same file. These outputs are generated, respectively, as a CAD part file, a triangulated surface mesh file in Cart3D and other formats, and ASCII keyworded results file. The geometric measurements that appear in the results file may include important reference quantities as projected area or overall length. Actors working with ComGeom2 include the ComGeom2 “user” and the “SCL administrator”. The ComGeom2 user configures new systems using the iLVL, and drawing on existing SCL components, generates the configured geometry using ComGeom2. In support of the user, the SCL administrator manages SCLs and their contents: administrators prepare new CAD parts and associated files for inclusion in existing SCLs as new SCL components. They also occasionally create completely new SCLs.

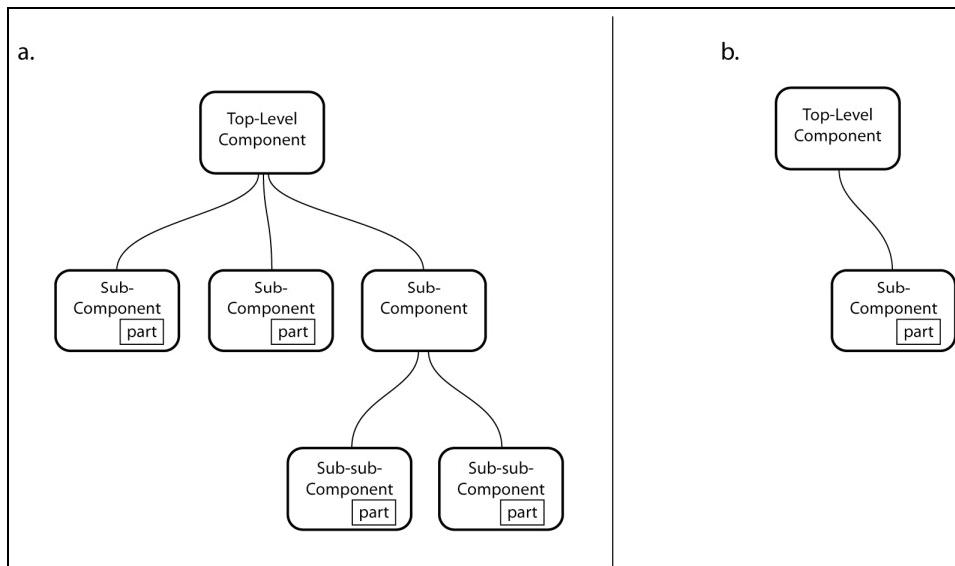


Figure 5: Geometry model component hierarchies in ComGeom2:
(a) general tree with multiple parts; (b) tree with single part.

An iLVL is an XML-based engineering system configuration document that follows the Launch Vehicle Language [13]. The ‘*i*’ in iLVL signifies that the document is an *instance* of LVL. Exploiting the nested markup-language syntax of XML, ComGeom2 users configure their geometric model by defining the relationship among its components in a tree data structure (Figure 5a). In the LVL framework, the overall model is represented at the tree’s root node by the top-level component. The top-level component may contain one or more sub-components, which may in turn contain additional sub-components. In ComGeom2, CAD parts are specified only at leaf nodes in the component tree. A model that contains only one part (*e.g.*, as in the LAS model) will still consist of two components due to the presence of system-wide parameters, such as the case name, that must be maintained at the top level (Figure 5b). The user therefore specifies in the iLVL the CAD parts that comprise the model and the parameters that dictate the size, shape, position, and orientation of each part (see Figure 6). Global mesh control parameters (triangle size and tolerances) are also chosen.

A system component library (SCL) is a repository of pre-defined, parameterized CAD parts. Each SCL part entry contains the files necessary for ComGeom2 to fully utilize the part. In addition to the part file, these files include an iLVL fragment, a face-tagging file, and a part formula file. An iLVL fragment is XML code corresponding to a component entry and contains the CAD part parameters that an SCL administrator might wish to expose to ComGeom2. An example is shown in Figure 6. A face-tagging file, unused in the present study, associates CAD surface elements with tags for application-specific boundary definition and labeling. The part formula file, also unused here, permits the SCL administrator to add mathematical expressions that transform parameters contained in the iLVL to those that appear in the CAD parts, usually to simplify the parameter interface exposed to the user. To establish relations of parameters in different parts of an assembly, the user may also create similar expressions in an *assembly* formula file (also unused here).

ComGeom2 input preparation may fall into one of three categories:

1. Adding a new part to an SCL;
2. Creating or modifying an iLVL’s part selection;
3. Modifying parameter values in an existing iLVL.

The first category is the most labor intensive, but also the least common, while the third is very simple and is repeated often. The LAS study required only one instance of the first type of input file preparation, followed by a few of the second and by over a thousand of the third type. The overhead associated with part library management is recouped as the level of re-use increases.

Preparing a new Pro/ENGINEER CAD part for inclusion in an SCL involves the following steps:

1. Ensuring that the geometric parameters of interest are properly defined in the CAD part;
2. Exposing these geometric parameters within the Pro/ENGINEER Pro/PROGRAM construct;
3. Adding to the part a reference coordinate system and including the position and orientation parameters (XPOS, YPOS, etc.);
4. Constructing an iLVL fragment file containing the part parameters to be exposed;
5. Creating the face-tagging file; and
6. Creating the part formulas file (optional).

The SCL itself follows a strict directory organization and file naming convention. Additional SCLs can be created as needed to organize parts.

To create an iLVL for a new geometric model, a user may insert iLVL fragments from an SCL into an existing iLVL and delete unneeded fragments. Only the parameters the user wishes to be exposed for modification need be included. Parameters that are not included will retain their original CAD part values.

Modifications to parameter settings in an existing iLVL typically occur either through a text editor or through the automation and integration framework provided by ModelCenter, which is described in Section 5. The ModelCenter based component wrappers we developed to integrate the iLVL file and ComGeom2 are flexible enough to accommodate without intervention the changes brought about by adding new parts to the SCL or reconfiguring the component layout in an iLVL.

```

<System>
<DryElements>
<DryElement name="Stage1" xsi:type="DryElementInstance">
  <Component name="las_plus_vehicle">
    <Geometry>
      <Variables>
        <Variable name="COMGEOM_CASE">LAS_1001</Variable>
        <Variable name="COMGEOM_MESHCONTROL">YES</Variable>
        <Variable name="COMGEOM_MESHCONTROL_MAXANGLE" units="deg">5</Variable>
        <Variable name="COMGEOM_MESHCONTROL_MAXDEVIATION_COEFF">0.001</Variable>
        <Variable name="COMGEOM_MESHCONTROL_MAXLENGTH_COEFF">0.005</Variable>
      </Variables>
    </Geometry>
    <SubComponents>
      <Component name="las_plus_comp">
        <Geometry>
          <Variables>
            <Variable name="AEEfileLocation">las_plus</Variable>
            <Variable name="XPOS" units="in">232.11</Variable>
            <Variable name="YPOS" units="in">0</Variable>
            <Variable name="ZPOS" units="in">0</Variable>
            <Variable name="XROT" units="deg">0</Variable>
            <Variable name="YROT" units="deg">0</Variable>
            <Variable name="ZROT" units="deg">0</Variable>
            <Variable name="LAS_FLARE_DIAMETER_RATIO">2</Variable>
            <Variable name="LAS_FLARE_ANGLE" units="deg">35</Variable>
            <Variable name="LAS_FLARE_LOCATION_RATIO">0.4</Variable>
            <Variable name="LAS_LENGTH" units="in">408</Variable>
            <Variable name="LAS_DIAMETER" units="in">36</Variable>
            <Variable name="ELLIPTIC_TIP">NO</Variable>
            <Variable name="LAS_TIP_FINENESS_RATIO">2</Variable>
            <Variable name="FLARE">YES</Variable>
            <Variable name="LAS_SHROUD_ANGLE" units="deg">32.5</Variable>
            <Variable name="LAS_SHROUD_EXIT_DIAMETER" units="in">202</Variable>
            <Variable name="LAS_SHROUD_LENGTH" units="in">145.32</Variable>
          </Variables>
        </Geometry>
      </Component>
    </SubComponents>
  </Component>
</DryElement>
</DryElements>
</System>

```

Figure 6: A typical ComGeom2 "iLVL" input file for the LAS model.

5. ModelCenter Integration for Multidisciplinary Studies

ModelCenter is a commercial framework for multidisciplinary analysis. Individual software tools are “wrapped” so that their inputs and outputs are exposed in the framework so that they are available for linking with other tools. When an analysis **A** produces data needed as input to analysis **B**, the outputs from **A** are linked to inputs for **B**, so that data transfer between the analyses is automated during execution. The ModelCenter environment provides convenient methods for specifying the relevant links. Individual software codes that have been wrapped for use in ModelCenter are called *Components*, while a collection of linked components is called a *Model*. Models can be used as Components in higher-level models, so that functionality can be managed in a hierarchy of arbitrary extent.

The ComGeom2 model used in this study includes 3 components, as shown in Figure 7. We chose to create a component to manage the iLVL input file (*iLVL-IN* in Figure 7), and a separate component to control execution of ComGeom2. A third component, *cbsetup*, is included to permit inspection of the mesh produced by ComGeom2.

The *iLVL_IN* component is an XML parser that is customized to interpret LVL syntax. It reads an iLVL input file, and exposes the names and values associated with “Variable” tags that lie between “Geometry” tags (which are shaded blue in Figure 6). These names and values are listed in the left pane of

Figure 7. The values can be updated through the ModelCenter interface and passed through links to be input to the ComGeom2 component.

The ComGeom2 component is a customized AnalysisServer wrapper written in Java. In addition to exposing command line arguments as inputs and dynamically parsing the results file for outputs in the ModelCenter interface, the ComGeom2 wrapper manages filesystem components necessary to store results. The critical feature that enables one to consider running large numbers of cases automatically is the wrapper's ability to dynamically create a results archive based on a case name specified in the input file.

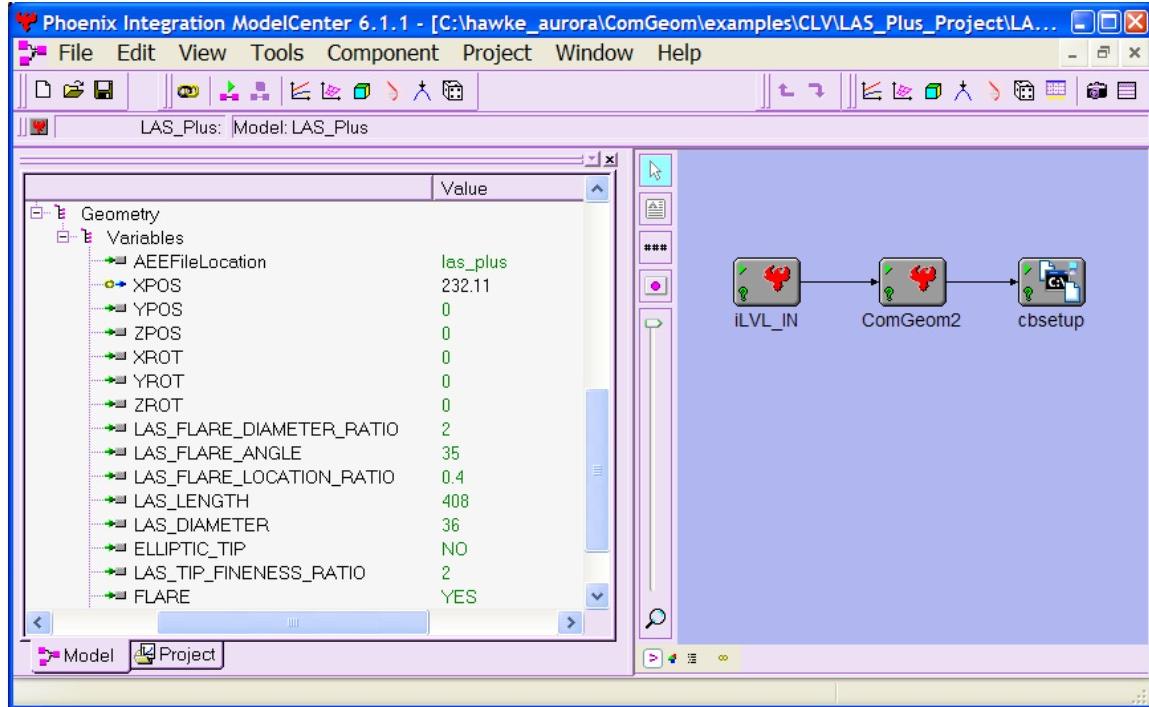


Figure 7: ComGeom2 in ModelCenter Environment

The ModelCenter application (as of version 6.1.1) provides a number of utilities that help users understand the sensitivity of system characteristics to changes in input parameter values. Parametric Survey (one-dimensional), Optimization (users can select from several algorithms), and Design of Experiments (which distributes analysis points in n -parameter space) are available for performing design studies. For the current application, which requires a full-factorial set of geometries, the use of DOE functionality is appropriate.

The set of parameters for the study and the values of each parameter need to be specified. The DOE tool provides an interface to generate regular n -dimensional arrays of points in the form of parameter scans, full factorial scans, fractional factorial scans, and other common automated DOE sets. It also permits the user to specify cases exhaustively in a comma-separated variable (CSV) format. In this study, the need to attach a unique identifier to each case rendered the CSV file-based approach the most appropriate. Furthermore, as a matter of practical convenience, the total set of cases was split into several smaller DOE runs in ModelCenter. The entire case set was tracked in a Microsoft Excel spreadsheet. As a result, the steps of setting up subsets in CSV format and configuring ModelCenter to execute DOE runs using these files as input were simplified. Details of case management are provided in Section 6.4.

6. Implementation for Launch Abort System Study

As specified in the geometric requirements, only the parameters for the LAS portion of the CLV (Crew Launch Vehicle) geometry were varied in this study. Geometry for a portion of the Service Module (SM) and upper stage was supplied to the CFD solvers because these features influence the flow being studied as well, but parameter values remained fixed throughout the study. The portion of the geometry aft of the LAS was constrained to remain in a fixed position to facilitate comparison of results for different geometry

instances by providing a common reference location. An equation was introduced to ensure that the aft portion remained fixed in space as the length of the LAS was varied. More details of the equation and its application are given below.

6.1 CAD Part Definition and Preparation for ComGeom2

We created a parameterized CAD part with the required LAS geometry using Pro/ENGINEER (Pro/E). Figure 8 shows the CAD model and part relations used for this task. We then prepared the Pro/E model for use in ComGeom2 following the steps outlined in section 4 and described in more detail in the *ComGeom2 System Component Library Administration Manual* [14]. We exposed part's parameters with Pro/PROGRAM, a function within Pro/E that allows external processes to update and vary parameters. We then created an iLVL input file (see Figure 6) that specifies the LAS part and lists its input parameters. Finally, we incorporated the iLVL into ModelCenter in the fashion described in Section 5.

About two days were needed to create the geometry with the driving parameters in Pro/E. In contrast, because the LAS study did not require tagged regions or formula files, preparing the geometry for ComGeom2 was brief, taking less than one hour.

Although we could have included the LAS part relations (Figure 8) in a ComGeom2 part formula file, for expediency we elected to code them directly into the Pro/E part. With more time, greater flexibility can be gained in using the ComGeom2's formula feature. For example, the formulas created for an initial part could be re-used and applied to subsequent similar parts within a design set. Further details of relation handling are included below in Section 6.4.1.

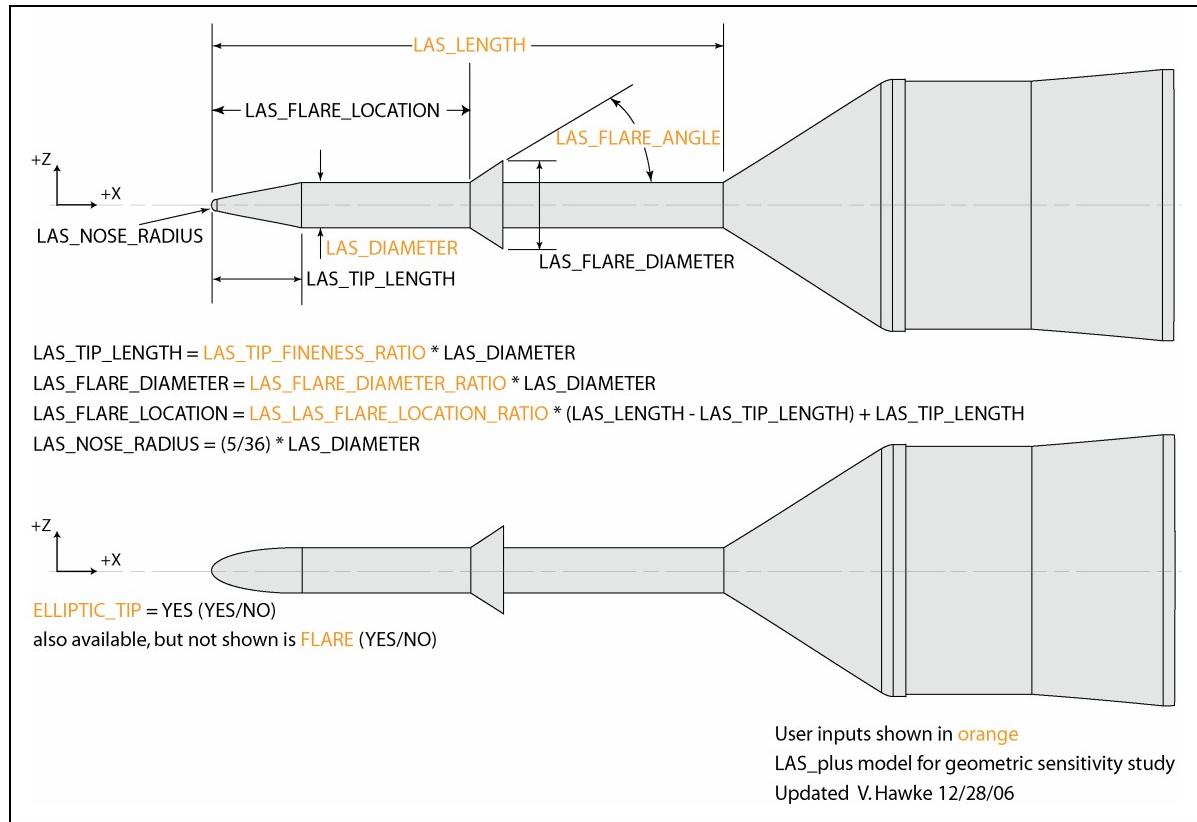


Figure 8: Parameterized LAS Geometry

6.2 Parametric Scope and Study Definition

Geometry parameter variations are shown below in Table 1. For the Cart3D runs, a full factorial set of cases was required. The flare-on case shape was driven by six parameters with three values per parameter possible giving a set of 3^6 , or 729, possible combinations for each of the possible two nose shapes. The flare-off cases were driven by only three parameters, giving 3^3 (27) variations for each of the two nose

shapes. This resulted in a full set of possible geometric variations of $2 \times (729 + 27)$, or 1512. Although we now know that the use of the ModelCenter DOE Tool to drive all LAS parameters from a CSV file is reliable, at the time of implementation we were uncertain in tool's ability to accommodate the YES/NO Boolean values of the FLARE and ELLIPTIC_TIP parameters. With no time to pretest the single set of cases based on the full automation of all (including the Boolean) parameters, we instead configured four sets of cases. For each set, we prepared a variant of the LAS CAD part that contained one of four possible unique combinations of nose and flare settings.

Table 1: List of LAS geometry parameter values

User inputs shown in orange		Baseline values underlined in blue		
#	Parameter	Description	Proposed Values	Units
1	LAS_FLARE_DIAMETER_RATIO	Flare diameter ratio .Ratio of Flare diameter over LAS diameter	1.5, <u>2.0</u> , 2.5	-
2	LAS_FLARE_ANGLE	Flare angle, base facing aft	25°, <u>35°</u> , 45	degrees
3	LAS_FLARE_LOCATION_RATIO	Forward most intersection on the flare with the LAS body Fraction of length along constant diameter section aft of nose	<u>0.4</u> , 0.6, 0.8	-
4	LAS_LENGTH	Length of tower forward of shroud, includes nose	326, <u>408</u> , 490	inches
5	LAS_DIAMETER	LAS tower diameter (same forward and aft of flare)	26, <u>36</u> , 46	inches
6	ELLiptic_TIP	LAS tip shape can either be the default sphere/cone or an ellipse	YES, <u>NO</u>	-
7	LAS_TIP_FINENESS_RATIO	Tip (nose) fineness ratio. Ratio of LAS tip length over LAS diameter	0.5, 1.25, <u>2.0</u>	-
8	FLARE	Allows flare to be toggled on and off	<u>YES</u> , NO	-
9	TOWER	Not yet available in the geometry	<u>YES</u> , NO	-

6.4 Study Implementation

6.4.1 Constraining Geometry Location

In the LAS geometry study, the use of mathematical expressions to impose constraints among various independent and dependent parameters was both convenient and necessary. In the ComGeom2-ModelCenter environment, it is possible to impose constraints between parameters in a number of locations:

1. Inside CAD parts, through *parameter relations*;
2. In ComGeom2, through *part* and *assembly formula files*; and
3. In ModelCenter, through *link formulas*.

In ComGeom2-ModelCenter framework, a constraint's most appropriate location depends on a) the generality and b) the accessibility needed in the constraint. Any universal constraints should be built into CAD parts and as such be relatively inaccessible to modification. Constraints pertaining to the general intent of a part when used with ComGeom2 should be captured by part formula files, which are maintained only by SCL administrators. Constraints that apply to a specific user's intent (especially across parts) are best suited for inclusion in assembly formula files, which are the purview of ComGeom2 users. If the user wishes to expose an application specific constraint in ModelCenter, links formulas are the most appropriate.

In LAS application, we built several convenient relations into the part, as listed in Figure 8. This step allowed us to recast several of the independent parameters as physically important nondimensional ratios. A necessary constraint, that arising from the requirement to locate the geometry in the x-direction (XPOS), was imposed with the LAS_LENGTH value using a ModelCenter link. An expression of the form

$$(1000 - (359.89 + <\text{prefix}>.\text{LAS_LENGTH})) \quad (1)$$

was built into the link, as shown in Figure 9. The amount 359.89 is the length in inches of the fixed portion of the geometry (CM + SM + fairing) and 1000 the required distance in inches from the global coordinate system to the aft end of the fairing. XPOS is a function of the location of the local nose coordinate system, which varies with the length of the LAS.

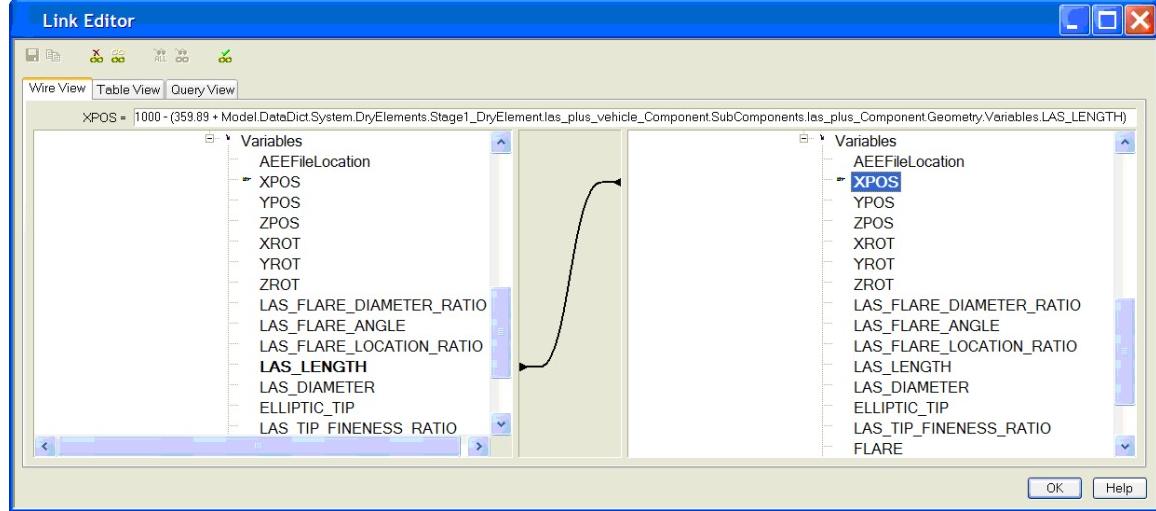


Figure 9: Linking in ModelCenter

LAS Study Case Summary Updated 4/28/06													
Green background = Case complete (X) Pink = Failed Geometry (F) Blue = Duplicate Case (D) Blue text = Baseline values, other values in orange Cases complete: 99.01 % Yellow = Problem Case (P) (see notes)													
Case Name	LAS_FLARE_DIAMETER_RATIO	LAS_FLARE_ANGLE (deg)	LAS_FLARE_LOCATION_RATIO	LAS_LENGTH	LAS_DIAMETER	LAS_TIP_FINENESS_RATIO	ELLIPTIC_TIP	FLARE	TOWER	Case Status (X = Complete)	To be run in Overflow (X = Yes)	Target	Notes
LAS_1001	2.0	35	0.4	408	36	2.00	NO	YES	YES	X	X	1	
LAS_1002	2.0	35	0.6	408	36	2.00	NO	YES	YES	X		1	
LAS_1003	2.0	35	0.8	408	36	2.00	NO	YES	YES	X		1	
LAS_1004	2.0	35	0.4	408	36	2.00	YES	YES	YES	X		1	
LAS_1005	2.0	35	0.6	408	36	2.00	YES	YES	YES	X		1	
LAS_1006	2.0	35	0.8	408	36	2.00	YES	YES	YES	X		1	
LAS_1007	-	-	-	408	36	2.00	NO	NO	YES	X		1	
LAS_1008	-	-	-	408	36	2.00	YES	NO	YES	X		1	
LAS_1009	1.5	35	0.4	408	36	2.00	NO	YES	YES	X		1	
LAS_1010	2.5	35	0.4	408	36	2.00	NO	YES	YES	X		1	
LAS_1011	2.0	25	0.4	408	36	2.00	NO	YES	YES	X		1	
LAS_1012	2.0	45	0.4	408	36	2.00	NO	YES	YES	X		1	
LAS_1013	2.0	35	0.4	326	36	2.00	NO	YES	YES	X		1	
LAS_1014	2.0	35	0.4	490	36	2.00	NO	YES	YES	X		1	
LAS_1015	2.0	35	0.4	408	26	2.00	NO	YES	YES	X		1	
LAS_1016	2.0	35	0.4	408	46	2.00	NO	YES	YES	X		1	
LAS_1017	2.0	35	0.4	408	36	0.50	NO	YES	YES	X			

Figure 10: Case summary spreadsheet (Microsoft Excel).

LAS_1001,2,35,0.4,408,36,2
LAS_1002,2,35,0.6,408,36,2
LAS_1003,2,35,0.8,408,36,2

Figure 11: Sample CSV text file input for DOE Tool.

6.4.2 Case Summary Spreadsheet and DOE Tool Input File

An important element in managing the large number of cases was the use of an Excel spreadsheet to track and summarize the runs (Figure 10). CSV files used to drive the ModelCenter DOE tool were produced directly from the spreadsheet, a significant labor saving step. The columns of the spreadsheet delineate the parameters, whereas the rows contain the parameter values of the cases. Subsets of the cases are viewed in the spreadsheet through the use of parameter value filters, which are activated with pull-down menus in each column. Through the use of these filters, the subset of cases corresponding to, for example, a specific nose shape and flare setting can be readily isolated, copied to a new worksheet, and saved as a CSV file (Figure 11).

6.4.3 ModelCenter Control of Parametric Variations

We configured ModelCenter to run large numbers of cases using the Design Of Experiments (DOE) tool. In Figure 12 we show the “Variables” tab of the DOE Tool interface, where we have entered six LAS parameters as design variables. A seventh parameter, the case name (COMGEOM_CASE) is also entered as a design variable so that ComGeom2 can automatically archive the results under a unique name. All of these parameters must appear in the iLVL. The DOE Tool requires that at least one output parameter be entered as a “Response”. We entered the mesh element count, although this result is not critical to our study. In Figure 13 we show the “Design Table” tab in the DOE Tool. This tab provides the interface for importing the CSV files we use to define our designs of experiments. The interface displays the cases in tabular format. Once the DOE is configured, we use the tool to launch it and monitor its progress.

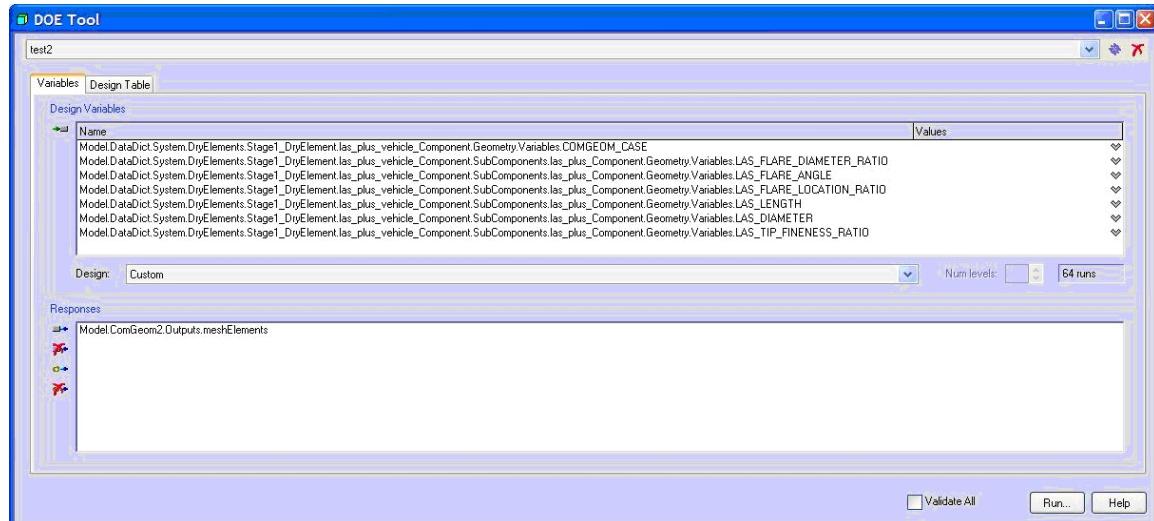


Figure 12: DOE Tool dialog window, Variables tab in ModelCenter.

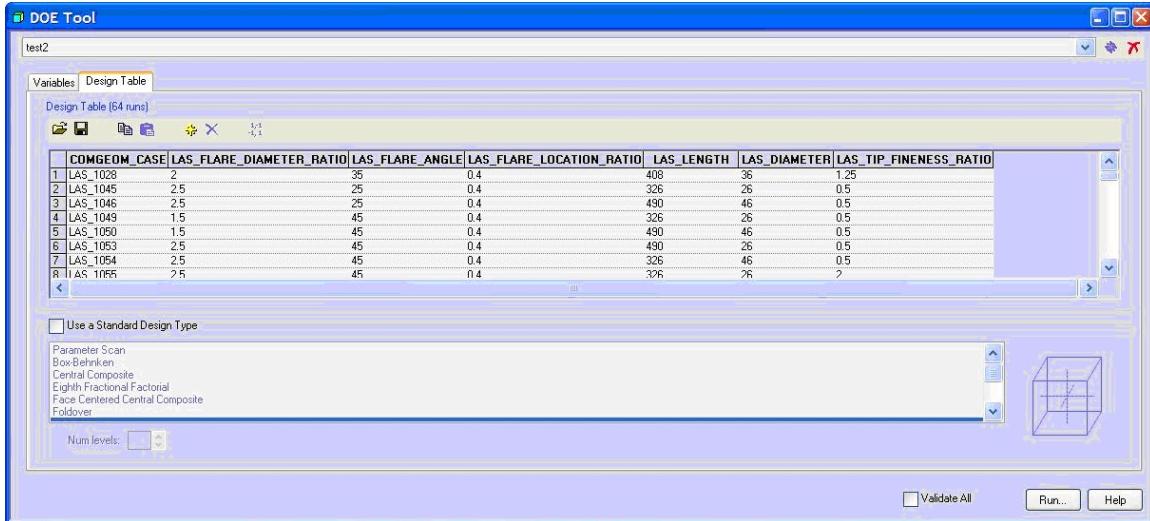


Figure 13: DOE Tool dialog window, Design Table tab in ModelCenter.

7. System Performance, Output, and Quality Checking

7.1 System Performance

While we were not allowed the time to fully optimize our approach to processing large sets of computational geometry, we took steps to monitor resource use and throughput to provide a basis for future improvement. Nor were our resources fully dedicated to this effort. As a result, the system performance we report was affected by an initial learning curve, a setup phase, a computer resource that was split with other work, and a somewhat crude timing method. Still, the results we obtained demonstrate that our approach is fundamentally sound and capable of highly sustained throughput.

The complete set of required cases were run over an eleven day period on a Dell M70 notebook computer with a 2.26 GHz Pentium M processor and 2 GB of RAM under Windows XP Professional SP2. The computer was exclusively dedicated to this effort only after business hours. At other times, the geometry processing competed with normal computer use for processor cycles.

We tracked the completion time of each case by recording the modification times of case output files, which the operating system displays to the nearest minute. In Figure 14 we show the progress of cases completed as a function of project time elapsed. In the first few days, when the project was still in ‘set-up’ phase, very few cases were run. Once the process was tested successfully, progressively larger sets of cases were run in ‘production’ mode, and throughput increased. The best performance of 5.5 minutes per case was seen over the final weekend when over 700 cases were run via the ModelCenter DOE tool. Based on this final throughput, one can determine that with enough storage space on a similar but dedicated machine, all of the 1500+ cases could be run as a single set in 5.78 days, or roughly half the actual time. However, even without the learning curve, we expect large production runs to require some front-end setup time.

To obtain a baseline sense of efficiency achieved by our ComGeom2-based automated approach, we compared our automated throughput to the time an experienced user would need to complete a single LAS case in Pro/E *manually* (*i.e.*, sans ComGeom2 and CAPRI). Given more time, a better standard for comparison would have been to automate Pro/E through “trail file” scripting. In our efficiency comparison, the user was forced to invoke Pro/MESH within Pro/E to produce triangulated surface meshes, which we required to contain the same number of triangles as were contained in the CAPRI meshes. We found that a user requires a total of just over one minute to change the parameters in Pro/E, regenerate the geometry, create and save the mesh with a unique file name, and archive all the files in a dedicated directory. During this time, the user is idle for only about 10 seconds (during grid generation). We conclude that while our ComGeom2-based approach is not efficient at the single case level, its full automation compensates for its inefficiency; the throughput achieved manually is unlikely to be sustainable for 1500+ cases with the needed reliability.

The approximately five-fold speedup that was achieved through the Pro/E with Pro/MESH approach warrants some discussion. To better understand the inefficiency, we examined how time was spent in the ComGeom2-based approach. We found that the bulk of the time spent per case is in CAPRI surface mesh generation. A case at the required mesh size of over 100,000 triangles consumed about six minutes, whereas the same case at 1,200 triangles finished in only two minutes. (A mesh with only 1,200 triangles will not produce a surface with sufficient surface definition for use in aerodynamic analysis.) The four-minute difference defines a lower bound on the time to mesh 100,000 triangles, suggesting that Pro/MESH, which required only about 10 seconds, is on the order of 20 times faster than the CAPRI mesher in producing meshes of the same size. Any speedup in the ComGeom2-based approach will require improvements in CAPRI's meshing efficiency. ComGeom2 is not currently configured to support Pro/MESH.

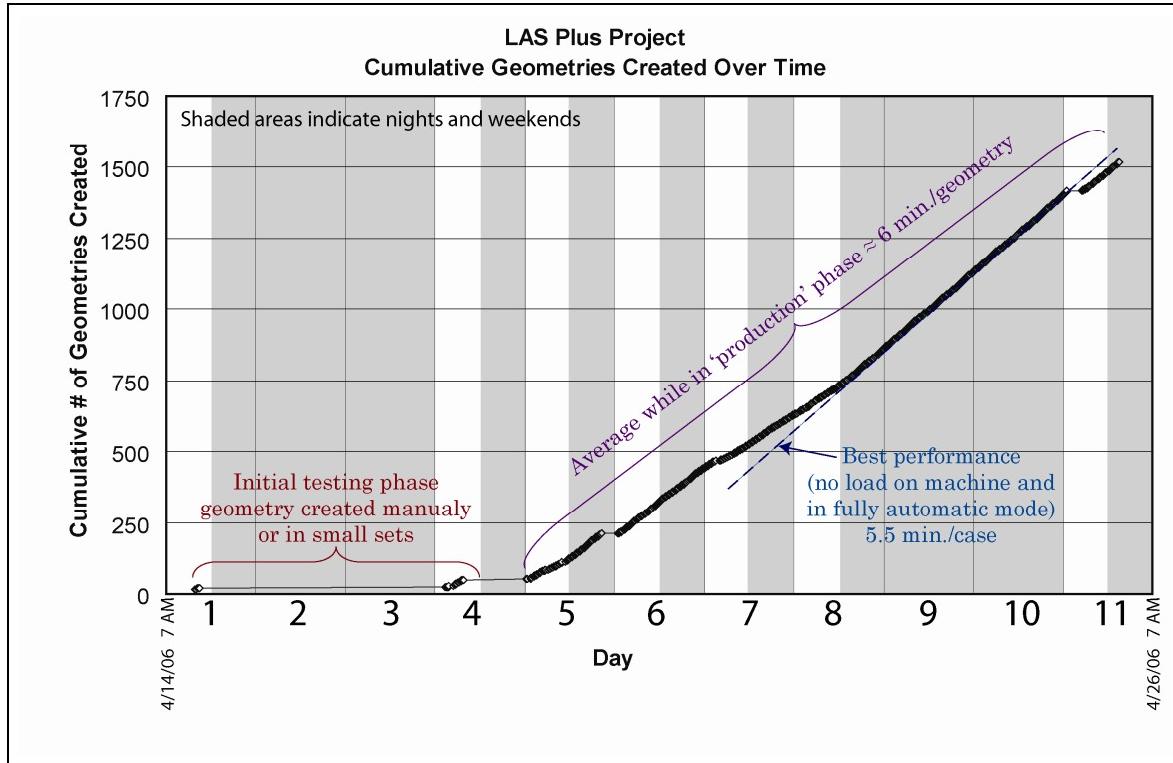


Figure 14: Cumulative geometries created over time

7.2 Case Preparation Issues

The truncated schedule led us to run some of the priority cases out of case name order and, as a side-effect, to unintentionally integrate duplicate cases into the summary list. Although we attempted to eliminate the redundancies, time constraints prevented a complete purge. In the end, the ModelCenter DOE Tool ran 23 duplicate cases. The impact on post processing time, however, was minimal. A fully automated run submission system, in which all of the cases are set up in one CSV file, would eliminate this problem in future applications.

7.3 LAS Geometry Quality Control

We inspected the LAS geometry for quality periodically on random cases as they completed. We carried out spot checks on output geometry to verify that the parameter values were set properly, and we monitored directory listings to locate failed cases. About one in every 50 to 75 cases was checked.

We carried out the parameter checks qualitatively by examining the selected models in *cbsetup*, a surface-mesh viewing tool usually associated with the panel-based flow solver, CBAero [15], and more rigorously by examining CAD part parameter values directly in Pro/E.

Because we configured the ComGeom2 wrapper to ignore ComGeom2 errors, ModelCenter was able to complete the valid cases in all designs of experiments despite occasional failures in ComGeom2. As a result, case failures did not impact case throughput, and we could search for failures at our leisure.

We located failed cases by examining the archive directories of each case. The mesh and other output files of such cases were either missing or empty. Once we understood the reasons for failure, we were able to anticipate and quickly locate other failed cases. The reasons for failure of a case include:

1. Failed Geometry, due to requested parameters defining non-physical geometry;
2. Unavailability of the Pro/E or CAPRI license during the few minutes that Pro/E was running for each case; and
3. System failure, such as lack of hard drive storage space.

Seventeen cases failed due to geometry interference (“Failed Geometry”, Category 1 above) when the LAS flare became large enough and was so far aft that it interfered with the shroud geometry. This failure is an instance of invalid regeneration in Pro/E. It is caught by ComGeom2 as a CAPRI exception, which allows ComGeom2 to shut down gracefully. Figure 15 shows a section used in creating a body of revolution that would result in a failed geometry. We could have avoided this critical fault by defining the LAS CAD part differently, *e.g.*, by creating separate sections for the shroud and the tower. The cases were not recreated later with a different part definition as they were deemed too extreme to be of interest for the sensitivity study, once they were discovered.

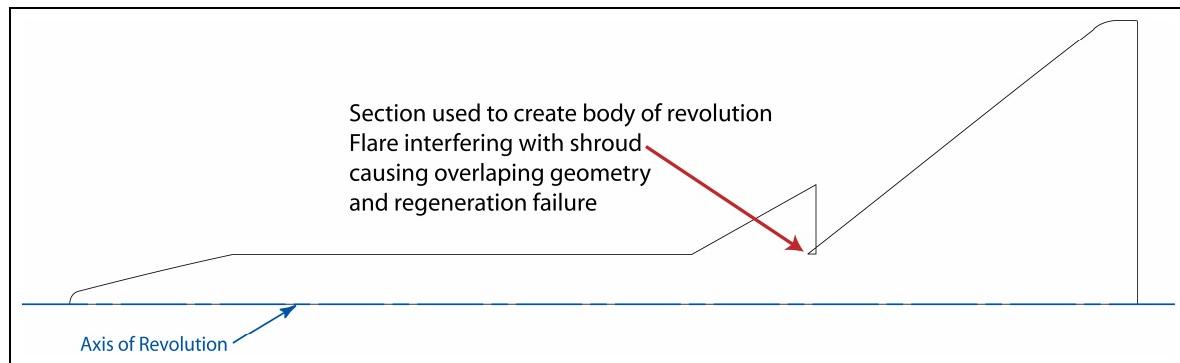


Figure 15: Example of Failed Geometry

When failures of Category 2 or 3 occurred, our most common response was to create a new CSV file containing the failed cases and rerun it in ModelCenter. License unavailability is typical in network floating license environments when the demand for the software license exceeds the number of licensed software seats. One may check for license availability before each ComGeom2 launch, but the time required to check may exceed the time required to re-submit the small number of jobs that fail due to license usage. A decision on automated license checking should be made on a case-by-case basis, depending on the local pattern of license usage. Similarly, automated checking of storage space requirements can be implemented where storage is a constrained resource.

7.4 Mesh Generation, Tailoring, and Quality Assessment

As we discussed in Section 3, the Cart3D analysts required LAS surface meshes, whereas the Overflow grid analyst required CAD parts that contain a defining section. Consequently, the process of supplying surfaces with qualities that met the requirements of the respective analysts differed in the two sets of cases.

The process of defining the mesh tolerances was iterative. In the CAPRI meshing framework, the mesh tolerances are defined globally as the maximum triangle size, the maximum angle between neighboring triangles, and the maximum departure of a triangle from the analytical surface. We selected a sphere-cone nose, flare-on geometry with the base-line parameter settings as the reference LAS configuration. We then produced the first candidate mesh using the default CAPRI mesher settings. The Cart3D analyst, viewing with *cbsetup* critical areas such as the nose tip, the flare, and the boost-protective-cover/service-module overlap, recommended grid refinements, which we carried out by gradually tightening the tolerance parameters. After several cycles, we obtained a surface that satisfied the analyst. The baseline mesh

consisted of about 113,400 triangles (Figure 16). We retained these mesh tolerances throughout the design of experiments cases.

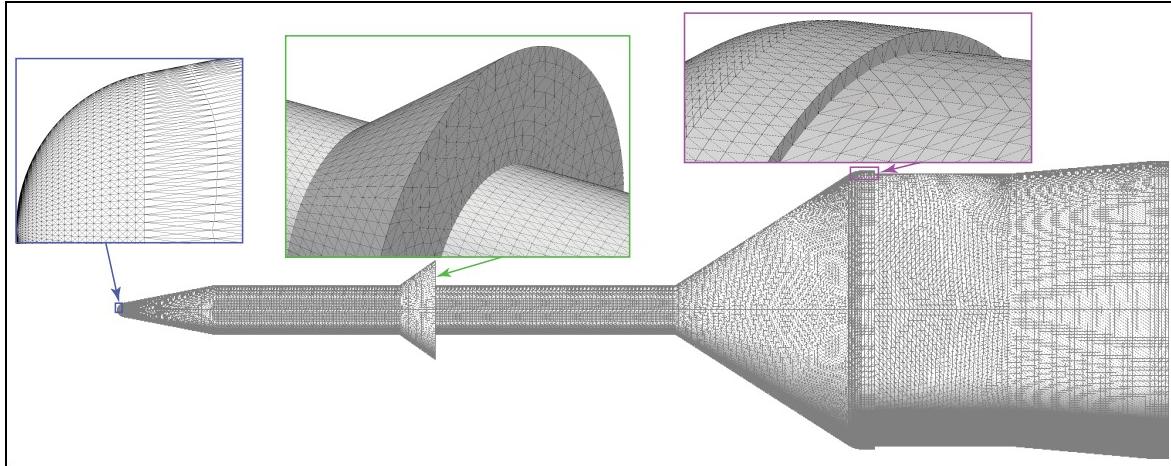


Figure 16: CAPRI mesh on LAS geometry

An additional step was needed to create geometry suitable for the script files used by Overgrid to create volume grids for Overflow. The LAS CAD part used in the Cart3D cases did not contain the generator curve needed by the Chimera Grid Tools scripts to produce the axisymmetric body meshes for Overflow. As there were only 85 Overflow cases, we chose to retain the original part and post-process it manually with a Pro/E trail file script rather than to create a new part. A trail file is generated as a by-product of an interactive Pro/E session, wherein the user's operations are recorded and can be replayed. In one such session, we opened a part newly regenerated by ComGeom2, activated cutting planes, and quartered the part, as shown in Figure 17. The process of invoking the trail file to quarter the part requires about one minute. Adding these cutting planes as a feature activated by a Boolean parameter in the CAD part, and including the part in the ModelCenter DOE at the parameter settings for which the Overflow geometry was needed, would have increased automation and decreased overall time consumed. However, at the time, we had little experience, and therefore no confidence, in using Boolean parameters in ModelCenter.

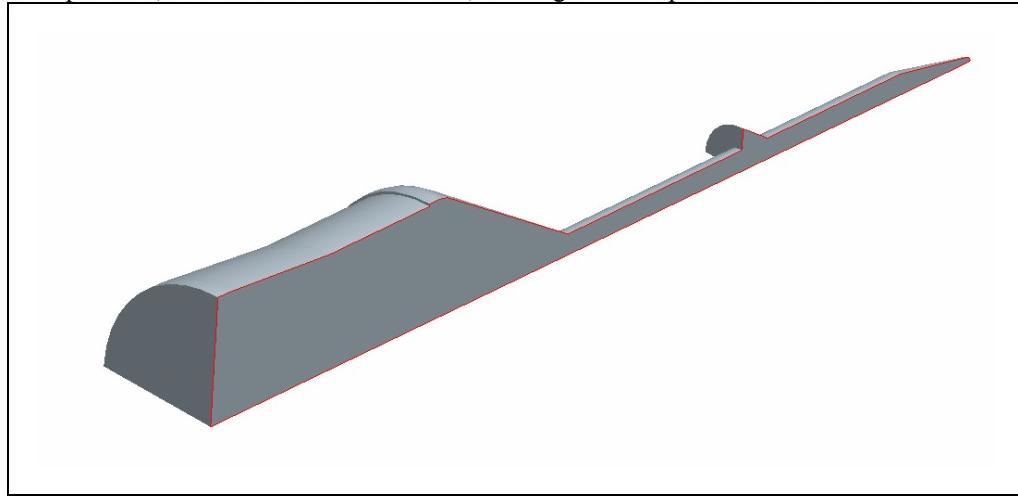


Figure 17: Quarter model for Overflow shown in Pro/E.

8. Conclusions

The curve in Figure 14 is characteristic of geometry generation with this tool, although the time-scale is application-dependent. There is preparation and customization effort that may take several days before a very large number of geometry instances can be generated automatically. If the entire project duration is to be less than a week, we recommend that one generate a few geometry instances interactively.

It is important that geometry specialists interact with the study leader and the analysts responsible for downstream tools when they build CAD parts for CAE applications. Iterative refinement should be expected, as all participants evolve their understanding of specific problem needs. ComGeom2 provides a formal process and a repository of parts, which supports traceability of design effort after the project is completed.

The level of automation should be matched to the operating environment. For projects of moderate scope, and where software license and execution resources are not expected to impose serious constraints, some user intervention and re-work is likely to be more efficient than development of customized scripts to achieve comprehensive process automation.

ComGeom2 efficiently generated the required geometry in this application. The additional cost of preparing the geometry for ComGeom2 was small compared to the time required to create the initial parameterized part. Some limitations in the specific CAD parameterization were noted, but are easily correctable. Improvement can be achieved in the integration with Overflow, so that generator curves are produced more efficiently, but efficient delivery of geometry customized for downstream tools was demonstrated. More extensive applications of ComGeom2 to mission-relevant design studies are planned.

Acknowledgements

The work described in this paper was carried out under the NASA Contract Number NNA 05BF35C. The authors would like to acknowledge the technical assistance and proof-reading provided by John Melton, Susan Cliff, and Steve Smith. We would also like to thank the Cart3D and Overflow analysis teams including Pieter Bunning and William Chan. Inquires into the use of ComGeom2 by individuals and organizations outside of NASA Ames Research Center should be directed to Robin M. Orans, Technology and Software Release Partnership Manager, Technology Partnerships, NASA Ames Research Center, MS 202A-3, Moffett Field, CA 94035-1000, <http://technology.arc.nasa.gov>.

References

- ¹ Hawke, V., Gage, P., Huyhn, L., and McGuire, K. "ComGeom: A Representation and Data Sharing Tool for Analysis of Launch Vehicles," *43rd AIAA Aerospace Sciences Meeting and Exhibit*, Reno, NV, AIAA-2005-1157, January 2005.
- ² <http://www.ptc.com>.
- ³ Manning, T. A., Gage, P., Nguyen, J. M., Haimes, R., "ComGeom2: A Geometry Tool for Multidisciplinary Analysis and Data Sharing," AIAA-2004-4303, *10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Albany, NY, August 30 - September. 1, 2004.
- ⁴ Malone, B., Papay, M, "ModelCenter: An Integration Environment for Simulation Based Design," *Simulation Interoperability Workshop*, Orlando, FL, March 1999.
- ⁵ Huebner, L. D., Hall, R. M., Haynes, D. A., Pamadi, B. N., Taylor, T. L., Seaford, C. M., "Status, Plans, and Initial Results for Ares I Crew Launch Vehicle Aerodynamics," IAC-06-D2.7.03, *57th International Astronautical Congress*, Velencia, Spain, October 2006.
- ⁶ Townsend, N.A., "Apollo Experience Report – Launch Escape Propulsion Subsystem", NASA TN D-7083, 1973.
- ⁷ Dotts, R.L., "Apollo Experience Report – Spacecraft Heating Environment and Thermal Protection for Launch Through the Atmosphere of the Earth", NASA TN D-7085, 1973.
- ⁸ NASA's Exploration Systems Architecture Study, NASA-TM-2005-214062, November 2005.
- ⁹ Aftosmis, M., "A parallel multilevel method for adaptively refined Cartesian grids with embedded boundaries," AIAA 2000-0808, *38th AIAA Aerospace Sciences Meeting and Exhibit*, Reno, NV, January 2000.
- ¹⁰ Buning, P. G., Jespersen, D. C., Pulliam, T. H., Klopfer, G. H., Chan, W. M., Slotnick, J. P., Krist, S. E., and Renze, K. J., *Overflow Users Manual*, Version 1.8aa NASA Langley Research Center, Hampton, VA, 2003.
- ¹¹ Chan, William M., Rogers, Stuart E., Nash, Steven M., Bunning, Pieter G., *Users Manual for Chimera Grid Tools*, Version 1.1, Released November 1999, <http://halfdome.arc.nasa.gov/cfd/CFD4/CGT/man.html>.
- ¹² Haimes, R., and Follen, G., "Computational Analysis Programming Interface," *Proceedings of the 6th International Conference on Numerical Grid Generation in Computational Field Simulations*, University of Greenwich, U.K., June, 1998.
- ¹³ Vander Kam, J. C., Gage, P. J., "The Launch Vehicle Language (LVL) Data Model for Evaluating Reusable Launch Vehicle Concepts," AIAA 2003-1330, *41st AIAA Aerospace Sciences Meeting and Exhibit*, Reno, NV, January 2003.
- ¹⁴ Manning, T., *ComGeom2 System Component Library Administration Manual*, unpublished, NASA Ames Research Center, 2005.
- ¹⁵ Kinney, D.J., "Aero-Thermodynamics for Conceptual Design", AIAA Paper AIAA 2004-31, *42nd AIAA Aerospace Sciences Meeting and Exhibit*, January 2004, Reno Nevada.